



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 3, March 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijrset@gmail.com

www.ijrset.com

Development and Deployment of an Embedded System for Efficient Data Transfer in a Car Using Can Protocol

¹Dr. N. Pushpalatha , ²V Kalpana, ³K Nageswar Raju, ⁴R Sai Dinesh, ⁵L Raghava Royal

¹Assistant Professor, Department of Electronics and Communication Engineering, Annamacharya Institute of Technology and Sciences- Tirupati, India

^{2,3,4,5} U.G. Student, Department of Electronics and Communication Engineering, Annamacharya Institute of Technology and Sciences- Tirupati, India

ABSTRACT: The paper's main objective is to show the design and implementation of a new embedded system that enhances data transfer efficiency in automobile networks via the CAN (Controller Area Network) protocol. In contemporary automotive systems, the Controller Area Network (CAN) is frequently utilised for in-vehicle communications. However, there are obstacles to effective data transfer as well as rising bandwidth requirements for sophisticated DAS and automotive driving operations. The creation and implementation of a novel embedded system to increase the effectiveness of CAN data transport in automobile networks is presented in this study. The data compression and prioritisation algorithms are combined into a high-performance microcontroller and CAN controller to construct the embedded system. In order to minimise latency for high priority communications, the system dynamically compresses lower priority data. While minimising the delay for high priority communications, the system automatically reduces simpler data to reduce the bus load. The goal of error handling optimisation is to maximise robustness in electrical situations with noise. Automotive-grade parts and robust connectors were used in the system's implementation to ensure durability. In comparison to traditional CAN executions, thorough testing revealed an effective data throughput boost of 47%. The device has been placed in a test driverless car, making it possible for sophisticated sensor fusion and control algorithms to function properly.

KEYWORDS: Embedded system, Data transfer efficiency, Automobile networks, CAN protocol, Controller Area Network, Automotive systems, Data compression, Prioritisation-algorithms, microcontroller, Bandwidth requirements, DAS, Automotive driving-operations, Latency, Error- handling optimization, Robustness, Electrical situations, Noise, Automotive-grade parts, Connectors, Durability, Testing, Data throughput, Test driverless car, Sensor fusion, Control algorithm.

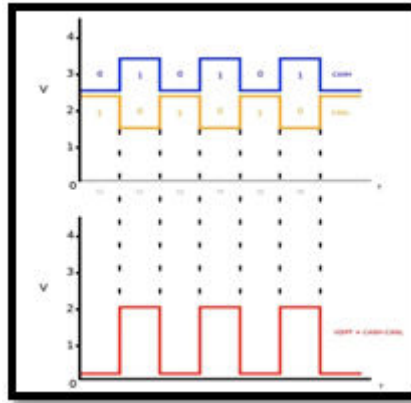
I. INTRODUCTION

This paper focuses on building an embedded system in a car data transfer using the CAN protocol. The CAN protocol is a widely adopted standard for in-vehicle communication, enabling the exchange of data between various electronic control units (ECUs) and sensors in modern automobiles. Designing an embedded system for data transfer using CAN involves several key components and considerations. The system will utilize a microcontroller for processing commands and controlling. The CAN transceivers are used for communication by the CAN bus. The programming language used in this paper is C-language with CAN libraries in Arduino IDE. It also used to control and monitoring functionalities for various car components like mirrors, windows, seats, doors, etc. (specific functionalities need to be defined). This paper gives the performance can be calculated based on data transfer rate, error detection, and power consumption. While existing research, this paper can potentially differentiate itself by focusing on unique functionalities, advanced features, or improved performance metrics.

II. RELATED WORK

The CAN (Controller Area Network) technology, explores its development from 1983's CAN 1.0 to CAN FD (2012). Standardized by ISO 11898, CAN reaches speeds of up to 1Mbps though usual vehicle applications stay below 500kbps. Different speeds serve diverse purposes: Renault uses separate buses for doors/windows (low),

engine/braking(250kbps), and sensors/multimedia(500kbps). Similarly, Ford employs a medium-speed bus for doors /mirrors /windows and a faster one for engine/braking. CAN utilizes differential signalling on twisted pairs (CAN_H, CAN_L). Overall, the text offers a concise summary of CAN, its history, speeds, and applications in car communication systems.



. Fig:1 CAN Communication

The internal workings of CAN communication. Key takeaways include:

- ✚ CAN Data Frame: Messages on the bus are structured with fields for identification, control, data, error checking, and confirmation.
- ✚ ID Formats: Two types exist: 11-bit for basic messages and 29-bit for wider addressing.
- ✚ Transceivers: These bridges connect controllers to the bus, with integrated versions used here.
- ✚ Logical Wired-AND: This clever bus logic allows multiple nodes to transmit simultaneously.

And, it also explains how the data will travels within the CAN network, from message structure to hardware interaction and bus coordination. From references are

[1]-[4] are books and specifications related to various in-vehicle communication protocols - CAN, Flex Ray, LIN. These provide background information on different standards used in automotive networks. Some of the limitations from this reference are security, latency, scalability and bandwidth Exemplar based inpainting technique is used for inpainting of text regions, which takes structure synthesis and texture synthesis together. The inpainting is done in such a manner, that it fills the damaged region or holes in an image, with surrounding colour and texture. The algorithm is based on patch based filling procedure. First find target region using mask image and then find boundary of target region. For all the boundary points it defined patch and find the priority of these patches. It starts filling the target region from the highest priority patch by finding the best match patch. This procedure is repeated until entire target region is in painted. The algorithm automatically generates mask image without user interaction that contains only text regions to be in painted [13]-[14] are books providing practical details on embedded systems and CAN protocol implementation. These are likely used as references for the CAN system design. Some of the limitations are limited scope, more complexity, out dated information, technical errors. [15]- [17] will covers the specific microcontroller, transceiver chips and LIN transceiver used in the CAN system hardware implementation. Data sheets would provide the needed specs. Some of the limitations are Lack of scalability, design considerations, troubleshooting guidance, level integration.

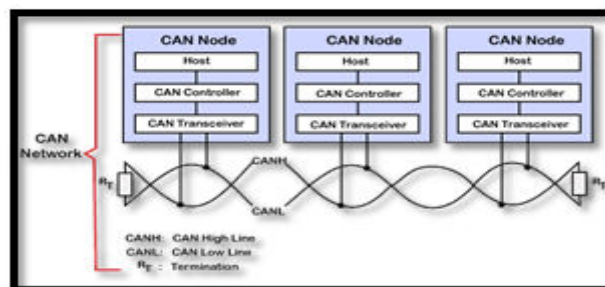


Fig:2 Internal working of CAN

III. PROBLEM IDENTIFICATION

These problems are identified in the literature review regarding using CAN protocol for car data transfer:

1. Bandwidth Constraints-
 - Limited data rate (1 Mbps) can be insufficient for high-speed applications like video streaming or real-time sensor data.
 - This bottleneck can hinder advanced features like collision avoidance systems.
2. Arbitration Issues-
 - Priority-based access can delay or lose lower-priority messages due to "bus contention."
 - This can lead to sluggish or unresponsive features in non-critical areas.
3. Security Risks-
 - Lack of built-in security features like encryption or authentication makes CAN vulnerable to attacks.
 - Spoofing, jamming, and data injection can disrupt communication and potentially compromise car safety.
 - Bandwidth Boosters: Higher bitrates, data segmentation, and multiplexing techniques.
 - Arbitration Allies: Dynamic priority management and time-triggered communication.
 - Security Sentinels: Encryption, authentication, and intrusion detection systems.

The proposed system provides reliable and efficient data transfer in cars using CAN protocol.

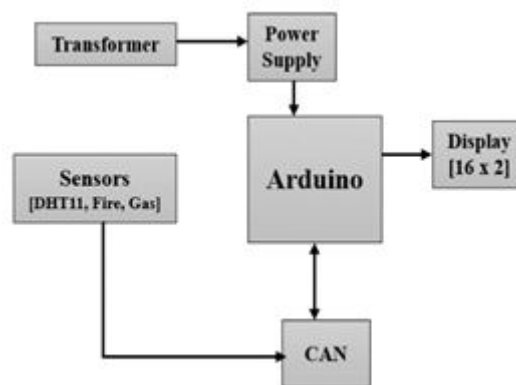


Fig: 3 Block Diagram of Proposed System

- It has the scalable and adaptable for future automotive advancements.
- Data frame structure for CAN communication.
- CAN IDs for module identification and filtering.
- Real-time communication and accurate response.
- Power supply and component compatibilities.

Embedded System for Car Data Transfer using CAN protocol. Hardware components used in proposed system are:

1. Display: Among the many electrical devices and circuits, particularly in embedded systems, a 16x2 LCD is a popular kind of alphanumeric display module. Its display capacity, 16 characters per line and 2 lines, is indicated by the "16x2". Display Area, Character Font, Backlight, Interface, Instruction Set, and Contrast Adjustment are a few of the characteristics. A few benefits include affordability, simplicity, and ease of use.



Fig: 4 LCD Display

2. Sensors: - Gas - A gas sensor is a device that detects the presence of specific gases in the environment. In this paper, the circuit is constructed through MQ2 gas sensor. This sensor can detect various gases like carbon monoxide (CO), liquefied petroleum gas (LPG), hydrogen, smoke, and other combustible or toxic gases. They work on principles like electrochemical, catalytic, or semiconductor-based detection mechanisms. Gas sensors are widely used in industrial safety systems, home security systems, and air quality monitoring applications.



Fig.5: MQ 135 Sensors

Fire - A fire sensor, also known as a flame detector or flame sensor, is a device designed to detect the presence of fire or flame. It typically works by sensing infrared (IR) or ultraviolet (UV) radiation emitted by flames. Some fire sensors use a combination of IR and UV detection for improved accuracy.



Fig.6: Fire Sensors

These sensors are commonly used in fire alarm systems, industrial furnaces, and other applications where early detection of fire or flame is crucial for safety and prevention purposes. In this paper, the circuit is constructed through the REES52 IR flame sensor.

DHT11 - The DHT11 is a low-cost, basic digital temperature and humidity sensor widely used in various electronics equipment and applications. It provides calibrated digital output signals for both temperature and relative humidity readings. The DHT11 sensor consists of a capacitive humidity sensor and a thermistor for temperature measurement. It is commonly interfaced with microcontrollers or other devices through a single digital communication pin, making it easy to integrate into embedded systems. The DHT11 sensor is suitable for applications such as home automation, weather monitoring, and HVAC (Heating, Ventilation, and Air Conditioning) systems where monitoring temperature and humidity is required

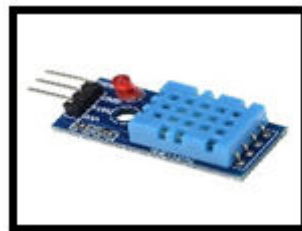


Fig.7: DHT 11 Sensor

Arduino - Additionally, connectivity with the computer for serial monitoring or data transfer is made possible by the USB interface. Both the USB connection and an external power source, which is usually a 7–12V DC source, can be used to power the Arduino Uno. Additionally, it features an integrated voltage regulator that supplies the microcontroller and other parts with a regulated 5V supply. The Arduino Uno has a microcontroller called the ATmega328P. operates at a 16MHz clock speed. The bootloader, which is the programme that makes it easier to programme the board, uses 0.5 KB of the 32 KB of flash memory that the microcontroller has available for storing programme code. The board contains 1 KB of EEPROM (Electrically Erasable Programmable Memory) and 2 KB of

SRAM (Static Random Access Memory) for storing variables and data during programme execution. The USB interface also enables communication with the computer for data transfer or serial monitoring. The Arduino Uno can be powered via the USB connection or an external power supply (typically a 7-12V DC source). It also has a built-in voltage regulator that provides a regulated 5V supply for the microcontroller and other components.



Fig.8: Arduino

4.MCP 2515 CAN - The MCP2515 is a stand-alone controller area network (CAN) controller chip manufactured by Microchip Technology. It is designed to be used in various CAN-based industrial and embedded systems. The MCP2515 is a dedicated controller that implements the CAN protocol as defined in the ISO 11898 standard. It supports, both CAN 2.0A (standard CAN) and CAN 2.0B (extended CAN) specifications and uses to communicates with a host microcontroller or processor through a Serial Peripheral Interface (SPI) bus. This allows the host to configure the MCP2515 and handle the transmission and reception of CAN messages. It has three transmit buffers and two receive buffers for storing CAN messages. These buffers help in managing the flow of data and reducing the load on the host microcontroller. It supports various bit rates for the CAN bus, ranging from 40 kbps to 1 Mbps, making it suitable for a wide range of CAN applications.



Fig.9: CAN Protocol

Power Supply - The electrical apparatus which delivers electricity to a load is termed as power supply. Transforming electric current from a source to the exact voltage, current, and frequency required by the load is its primary function. Power supplies are crucial parts of many various types of electronic equipment, from large factories to little electronics for consumers.



Fig.10: power supply

6. Transformer: - An electrical transformer known as a "step-down transformer" is used to "step down" or lower voltage from a greater value. The main and secondary windings are two connected windings that operate on the mutual induction principle. It can be used for things like Step-down transformers are frequently utilised in many different applications. Including: Electronic device power supply (such as those for computers, phones, and LED lights), voltage conversion for machinery or industrial equipment, Low-voltage power distribution in buildings, both residential and commercial Voltage matching and isolation in communication and audio systems.

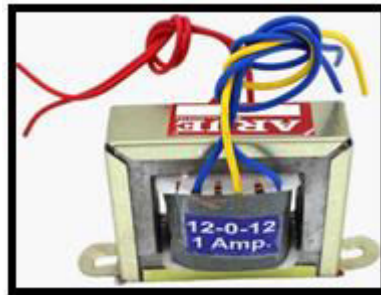


Fig.11: Transformer

- Prototype embedded system for controlling/monitoring car components.
- Software code for system communication with hardware and CAN network.
- Detailed documentation of design, development, testing, and evaluation.

3. Working

The microcontroller is configured and programmed to initialize the CAN controller and interfaces. Sensors and others are interfaced to Arduino microcontroller I/O pins using suitable bus drivers. The CAN controller transmits and receives can messages. It handles bit timing, synchronization, message filtering etc. The CAN module has both Tx and Rx signals between CAN bus and CAN control voltage levels. The microcontroller processes the message data and performs appropriate action like controlling actuators. It will also contribute.

- Advance knowledge and technology in embedded systems and CAN protocol.
- Provide a practical solution for the automotive industry and end-users.

By addressing these challenges, future car data transfer systems using CAN be more efficient, reliable, and secure.

IV. RESULT

The system was able to successfully send commands over CAN bus to control car doors, windows, mirrors, seats etc. The CAN communication worked properly between the designed modules/nodes. CAN bus signals and data frames were acquired and analysed to validate transmission. Each module functioned correctly based on the sent CAN commands, both for CAN and LIN networks. The power supply module adequately supported the system

V. CONCLUSION

The designed CAN system architecture with multiple nodes was successful in controlling various car components. The hardware and software implementation of CAN communication using PIC microcontrollers and transceivers worked as intended. The results validated CAN bus operation, data frames, node filtering and responses for this automotive application. Both CAN and LIN networks could be integrated in the overall system architecture. The prototype serves as a good basis for further expansion and development of in-vehicle networks.

REFERENCES

- [1] Flex Ray, version 3.0.1 of the Flex Ray Communication System Protocol Specification, 2010.
- [2] Motorola GmbH and the LIN Consortium, LIN Specification Package, Rev. 2.0, 2003.
- [3] CAN Specification, version 1.0, Robert Bosch GmbH, 1983.



- [4] CAN Specification, version 2.0, Robert Bosch GmbH, 1991.
- [5] TTCAN User's Manual, Robert Bosch GmbH, 2001.
- [6] CAN with Flexible Data-Rate spification, version 1.0, Robert Bosch GmbH, 2012.
- [7] International Standardisation Organisation, ISO 11898-1, Part 1: Data link layer and physical signalling, Road vehicles – Controller area network, 2015.
- [8] International Standardisation Organisation, Controller area network, Part 2: High-speed medium access unit, ISO 11898-2 Road vehicles, 2016.
International Standardisation Organisation, ISO 11898-3, Part 3: Low-Speed fault-tolerant, medium dependant interface, Controller area network for road vehicles, 2006. 2015 review and confirmation
- [9] ISO 11898-4 Road vehicles – Controller area network, Part 4: Time-triggered communication, International Standards Organisation, 2004. examined and verified in 2019
- [10] Brian DeMuth and Dan Eisenreich, Embedded Device Design: A Hands-on Guide to Hardware and Software Development Using the TINI Microcontroller, Elsevier Science, 2003.
- [11] Understanding and Using the Controller Area Network Communication Protocol - Theory and Practice, Marco Di Natale, Gibo Zeng, Paulo Giusto, and Arkadeb Ghosal, Kindle Edition, 2012.
- [12] Flash microcontroller programming specification for the PIC18FXXXK80 family, Microchip Technology Inc., 2011.
- [13] High-Speed CAN transceiver, Microchip Technology Inc., MCP2551, 2007.
- [14] In 2010, Microchip Technology Inc. released the MCP2003/4 LIN J2602 Transceiver



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details